-continued

```
        ColorSeqName [N-1] = valueN-1 [ (ColorNameN-1) ]
Stock image format:
    ImageName
        Format:         [Bitmap | Windows Metafile | JPEG]
Database connection format:
    ConnectionName
        Server:         Server
        Database:       Database
        User:           User
        Default schema:
Data source format:
    DataSourceName [ (DataType paraml, ...) ]
        Connection:     ConnectionName
        SQL:        SQL
Scene format:
    SceneName[ (DataType param1, ...) ]
        Local parameters:
          Parameter name:       ParamName
          Description:          Description
          Default value:            [Value]
          . . .
        Viewpoints:
          Viewpoint name:       ViewpointName [ (Built-in) ]
          Description:          Description
          Location:             (X, Y)
          Zoom factor:          Zoom %
          . . .
        Objects:
          ExtendedObjectName
          Properties:
          DataType PropertyName = PropertyValue
          . . .
          Events:
          EventName (DataType EventParam1, . . . )
              [Param = Expression]
              . . .
              [Action (Option1, . . . )]
        . . .
```

[0085] The pseudocode may be displayed using a color-coded syntax. For example section headings may be displayed in green, data types may be displayed in blue, and all other text may be displayed in black. The pseudocode may be updated each time a change is made anywhere within the world.

[0086] The Output window is a dockable control bar which displays compile and execution information as a world is being developed and viewed. It also displays warnings and error messages. The Output window may be resized vertically when docked and both vertically and horizontally when floating.

[0087] In runtime mode, a viewpoints dialog box may be provided for navigating a world. Scenes allowing direct access may be displayed in the Scenes combobox. The viewpoints for the selected scene may be displayed in the Viewpoints listbox. Selecting a viewpoint updates the values of the location edit controls. The values in these controls can be changed to a custom designation by specifying a location and zoom factor.

[0088] A Layout Wizard may be provided to guide the user through the steps in creating a new layout. The first two steps common to all layouts are to select the layout and then the data source. Each layout may then have as many additional layout-specific steps as necessary to create the layout.

[0089] Layouts are spatial arrangements of data elements resulting from a data source with supporting context information such as a set of axes. The layouts may fall into the following set of 5 categories:

[0090] Charts—used for presenting series information.

[0091] Maps—used for laying out multiple curves or series of information within a single data source.

[0092] Hierarchies—used for showing object dependencies.

[0093] Patterns—used for laying out independent objects in an ordered pattern.

[0094] Forms—used for displaying a single record in a data source (if the data source results in more than one row, subsequent rows are ignored).

[0095] Layouts can be placed anywhere on a scene or data element drawing. Some layout types such as maps and scatter charts support more than one data element, thus allowing "layering" of information. In this case, each data element can be assigned a visibility that depends on the current zoom level, thus allowing more information to be presented at greater zoom levels.

[0096] Axis nodes are attached to layouts that require them. For some chart types such as bar and column charts which have an axis that is based on text labels rather than numeric values, one or more of these axes may require a data source and may thus be data-bound. Axes that may be supported include linear, logarithmic, and date/time axes. These axes may also support auto-adjusting label and tick increments so that zooming may provide more information on precise axis values.

[0097] Certain layout types may use locators rather than axes to place data elements. These locators may not be displayed visually at design time, but their properties may be displayed in the Object Inspector along with those of the other objects in a layout. Composite layouts, where a data element within a layout may contain a layout itself, may also be used. Typically, the data source associated with the second layout may be parameterized, with one or more parameters linked to a column of the parent layout's data source.

[0098] Identifiers are names for elements within a virtual world such as objects, data sources, scenes, parameters, user classes, colormaps, color sequences, and viewpoints. Identifiers are case insensitive and may be of any length greater than or equal to one character.

[0099] Object properties consist of named attributes that define an object's appearance in terms of a functional expression. Access to these properties are provided through the Object Inspector. Their values are set at runtime based on the calculated result of the expressions. An expression may include the names of one or more data source columns which are automatically bound to a result row at runtime.

[0100] Object property expressions result in one of the following base data types: Boolean, Numeric, String, Point, PointList, and Image. Derived Numeric types include Color, DateTime, Enum, Integer, and Percentage. Derived String types include FilePath (or URL) and FontName.